Competição 1 - Predição de Cobertura de um Plano de Saúde

Profª. NADIA FELIX FELIPE DA SILVA
Discentes: Matheus Andrade Dutra e Mariana Inácia Xavier Borges

1. madutra@discente.ufg.br
2. mariana_borges@discente.ufg.br

Material elaborado em parceria com os professores NADIA FELIX FELIPE DA SILVA

2022

INF
INSTITUTO DE INFORMÁTICA

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

Sumário

INF
INSTITUTO DE INFORMÁTICA

Descrição do problema

Descrição do problema

Devido ao alto número de requisições em uma operadora de plano de saúde, encaminhadas para o setor administrativo e visando diminuir os gastos com a análise de dados, é necessário que se implemente um código capaz de fazer essa análise sem que um auditor precise investigar a fundo caso a caso.

Descrição do problema

Dados fornecidos:

- O conjunto de dados de treinamento fornecido contém 227.122 dados e 32 variáveis
- rótulo ”aprovar ou negar”
- ID do usuário
- 30 variáveis fornecidas:
    - 15 variáveis ou 50% delas não possuem Campos nulos.
    - Tempo de doença e sua unidade (dias, semanas, meses).
    - Tipo de doença 99% dos campos estão vazios.
    - Tipo de consulta contém 95% dos campos não usados.
    - O tipo de saída também era completamente nulo.

Descrição do conjunto de dados

Pré-processamentos utilizados

- Preenchimento dos campos nulos.
- StandardZation
- LabelEncoder
- OneHotEncoder

Código de pré-processamento

Código 1: Pré-processamento

```
from sklearn.preprocessing import StandardScaler, LabelEncoder, OneHotEncoder
import numpy as np

def standardScalerFunc(data):
  ss = StandardScaler()
  sstransformed = ss.fit_transform(data)

  return ss, pd.DataFrame(sstransformed)

def labelEncoderFunc(data):
  ss = LabelEncoder()
  sstransformed = ss.fit_transform(data)

  return ss, pd.DataFrame(sstransformed)

def oneHotEncoderFunc(data):
  ohe = OneHotEncoder(sparse=False, handle_unknown='ignore')
  ohetransformed = ohe.fit_transform(data)

  return ohe, pd.DataFrame(ohetransformed)
```

Algoritmos utilizados

Algoritmos utilizados

- Linguagem: Python 3.10;
- Bibliotecas:
  - numpy,
  - pandas,
  - os,
  - sklearn.preprocessing importado:
    - StandardScaler,
    - LabelEncoder,
    - OneHotEncoder.
- Foi utilizado o algoritmo RandomForestClassifier, afinal por ter uma coleção de dados para apontar uma decisão, este algoritmo seria o mais preparado para a tarefa de classificar a acurácia da cobertura do plano de saúde.
- A biblioteca fornecida pelo SKLearn foi muito útil, pois nos permitiu compartilhar a implementação da função train-test-split, baseada em dados de treinamento e teste. Os parâmetros que foram usados nesta função foram:
  - Tamanho do teste = 0,3, 30% para teste, 70% para treinamento,
  - Aleatório = Verdadeiro, assim tentando melhorar a distribuição de aleatoriedade nos dados.

Resultados

Resultados

Os resultados obtidos no teste foram de média satisfação, sendo que a acurácia obtida foi de 0.7147826086956521 ou 71\%, resultando no melhor resultado obtido no treino de 71\%. É necessário avaliar novamente algumas variáveis utilizadas para que se possa obter uma melhor exatidão.

	precision	recall	f1-score	support
Autorizado	0.74	0.90	0.81	30832
Negado	0.60	0.33	0.42	14593
accuracy			0.71	45425
macro avg	0.67	0.61	0.62	45425
weighted avg	0.69	0.71	0.69	45425

Table 1: Tabela de classificação

Obrigado!

Dúvidas ou sugestões:

madutra@discente.ufg.br ou
mariana_borges@discente.ufg.br

INF
INSTITUTO DE INFORMÁTICA

UFG
UNIVERSIDADE FEDERAL DE GOIÁS